

IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Original) A computer-implemented method for creating a graphical program, the method comprising:

creating a graphical user interface for the graphical program in response to user input;

displaying an event structure node in a block diagram for the graphical program in response to user input; and

configuring the event structure node to receive and respond to one or more user interface events during execution of the graphical program.

2. (Original) The method of claim 1,

wherein the event structure node comprises one or more sub-diagrams;

wherein said configuring the event structure node to receive and respond to the one or more user interface events comprises configuring the one or more sub-diagrams to receive and respond to the one or more user interface events.

3. (Original) The method of claim 2,

wherein, for each of the one or more sub-diagrams, said configuring the sub-diagram comprises:

specifying one or more user interface events to which the sub-diagram corresponds, in response to user input;

including graphical source code in the sub-diagram in response to user input, wherein the graphical source code is operable to respond to the one or more user interface events to which the sub-diagram corresponds.

4. (Original) The method of claim 3,

wherein said including graphical source code in the sub-diagram comprises including two or more interconnected nodes in the sub-diagram.

5. (Original) The method of claim 1,
wherein the block diagram comprises a data flow block diagram.

6. (Original) The method of claim 1, further comprising:
executing the graphical program;
wherein one or more user interface events to which the event structure node is configured to receive and respond are generated during execution of the graphical program;
wherein the event structure node is operable to receive and respond to the one or more user interface events generated during execution of the graphical program.

7. (Original) The method of claim 6,
wherein the one or more user interface events generated during execution of the graphical program are generated in response to user input to the graphical user interface of the graphical program.

8. (Original) The method of claim 6,
wherein, during execution of the graphical program, the block diagram executes on a first computer system and the graphical user interface is displayed on a display of a second computer system.

9. (Original) The method of claim 6,
wherein, during execution of the graphical program, the graphical user interface is displayed on a display of a computer system and the block diagram executes on a reconfigurable instrument connected to the computer system.

10. (Original) The method of claim 1,

wherein said configuring the event structure node to receive the one or more user interface events comprises configuring the event structure node to receive notification when the one or more user interface events are generated during execution of the graphical program.

11. (Original) The method of claim 1,

wherein said configuring the event structure node to receive the one or more user interface events comprises configuring the event structure node to receive information specifying occurrences of the one or more user interface events during execution of the graphical program.

12. (Original) The method of claim 1,

wherein said configuring the event structure node to receive and respond to one or more user interface events comprises displaying graphical source code in the event structure node operable to receive and respond to the one or more user interface events.

13. (Original) The method of claim 1,

wherein said configuring the event structure node to receive and respond to one or more user interface events comprises configuring the event structure node to receive and respond to a first user interface event;

wherein the first user interface event specifies a first user interface element of the graphical user interface and an action performed on the first user interface element.

14. (Original) The method of claim 13, wherein the first user interface element comprises one of:

- an indicator;
- a control;
- a menu element;
- a window.

15. (Original) The method of claim 1,

wherein said configuring the event structure node to receive and respond to one or more user interface events in response to user input comprises receiving user input via a graphical user interface dialog to specify the one or more user interface events.

16. (Original) The method of claim 1, further comprising:

displaying an event registration node in the block diagram in response to user input;

configuring the event registration node to dynamically register a first user interface event during execution of the graphical program;

wherein, after said dynamically registering the first user interface event, the event structure node is operable to receive and respond to the first user interface event.

17. (Original) The method of claim 16,

wherein said configuring the event registration node to dynamically register a first user interface event during execution of the graphical program comprises connecting the event registration node to the event structure node in response to user input.

18. (Original) The method of claim 1, further comprising:

displaying an event un-registration node in the block diagram in response to user input;

configuring the event un-registration node to dynamically un-register a first user interface event during execution of the graphical program;

wherein, after said dynamically un-registering the first user interface event, the event structure node does not receive and respond to the first user interface event.

19. (Original) A computer-implemented method for creating a graphical program, the method comprising:

displaying an event structure node in a block diagram for the graphical program in response to user input; and

configuring the event structure node to receive and respond to one or more programmatic events during execution of the graphical program.

20. (Original) The method of claim 19,
wherein the event structure node comprises one or more sub-diagrams;
wherein said configuring the event structure node to receive and respond to the one or more programmatic events comprises configuring the one or more sub-diagrams to receive and respond to the one or more programmatic events.

21. (Original) The method of claim 20,
wherein, for each of the one or more sub-diagrams, said configuring the sub-diagram comprises:
specifying one or more programmatic events to which the sub-diagram corresponds, in response to user input;
including graphical source code in the sub-diagram in response to user input, wherein the graphical source code is operable to respond to the one or more programmatic events to which the sub-diagram corresponds.

22. (Original) The method of claim 19, wherein the one or more programmatic events comprise one or more of:
a user interface event;
a system event;
a timer event;
an event generated in response to data acquired from a device.

23. (Original) A computer-implemented method for creating a graphical program, the method comprising:
creating a graphical user interface for the graphical program in response to user input;

creating a block diagram for the graphical program in response to user input; and
configuring the graphical program to receive and respond to one or more user
interface events in response to user input.

24. (Original) The method of claim 23,
wherein said creating the block diagram for the graphical program in response to
user input comprises arranging a plurality of nodes on a display and interconnecting the
plurality of nodes in response to user input.

25. (Original) The method of claim 23, further comprising:
executing the graphical program;
wherein one or more user interface events to which the graphical program is
configured to receive and respond are generated during execution of the graphical
program;
wherein the graphical program is operable to receive and respond to the one or
more user interface events generated during execution of the graphical program.

26. (Original) The method of claim 23,
wherein said configuring the graphical program to receive and respond to one or
more user interface events comprises configuring the block diagram to receive and
respond to the one or more user interface events.

27. (Original) The method of claim 23,
wherein said configuring the block diagram to receive and respond to one or more
user interface events comprises including graphical source code in the block diagram
operable to receive and respond to the one or more user interface events.

28. (Original) The method of claim 23,
wherein said configuring the graphical program to receive and respond to one or
more user interface events in response to user input comprises receiving user input via a
graphical user interface dialog to specify the one or more user interface events.

29. (Original) The method of claim 24,
wherein said configuring the graphical program to receive and respond to one or more user interface events in response to user input comprises including an event structure node in the block diagram in response to user input;
wherein the event structure node is operable to receive and respond to the one or more user interface events.

30. (Original) The method of claim 29,
wherein the event structure node includes one or more sub-diagrams;
wherein each sub-diagram includes graphical source code specifying a response to one or more user interface events.

31. (Original) A computer-implemented method for executing a graphical program, the method comprising;
creating the graphical program, wherein said creating the graphical program comprises configuring the graphical program to receive and respond to one or more user interface events;
executing the graphical program;
receiving user input causing generation of a first user interface event, wherein the graphical program is configured to receive and respond to the first user interface event;
and
sending the first user interface event to the graphical program.

32. (Original) A memory medium for creating a graphical program, the memory medium comprising program instructions executable to:
create a graphical user interface for the graphical program in response to user input;

display an event structure node in a block diagram for the graphical program in response to user input; and

configure the event structure node to receive and respond to one or more user interface events during execution of the graphical program.

33. (Original) The memory medium of claim 32,
wherein the event structure node comprises one or more sub-diagrams;
wherein said configuring the event structure node to receive and respond to the one or more user interface events comprises configuring the one or more sub-diagrams to receive and respond to the one or more user interface events.

34. (Original) The memory medium of claim 33,
wherein, for each of the one or more sub-diagrams, said configuring the sub-diagram comprises:

specifying one or more user interface events to which the sub-diagram corresponds, in response to user input;

including graphical source code in the sub-diagram in response to user input, wherein the graphical source code is operable to respond to the one or more user interface events to which the sub-diagram corresponds.

35. (Original) The memory medium of claim 34,
wherein said including graphical source code in the sub-diagram comprises including two or more interconnected nodes in the sub-diagram.

36. (Currently Amended) A computer-implemented method for creating a graphical program, the method comprising:

displaying a first node in a block diagram of the graphical program in response to user input; and

associating graphical source code with the first node in response to user input;

wherein the graphical source code is operable to execute in response to an user interface event.

37. (Currently Amended) The method of claim 36,
wherein said graphical source code executing in response to the user interface event comprises the graphical source code executing in response to the user interface event during execution of the graphical program, wherein the user interface event is generated during execution of the graphical program.

38. (Previously Presented) The method of claim 36,
wherein the graphical source code comprises a plurality of interconnected nodes.

39. (Previously Presented) The method of claim 38,
wherein said associating the graphical source code with the first node comprises:
displaying the plurality of nodes in response to user input; and
interconnecting the plurality of nodes in response to user input.

40. (Previously Presented) The method of claim 36,
wherein said associating the graphical source code with the first node in response to user input comprises displaying the graphical source code within the first node in response to user input.

41. (Currently Amended) The method of claim 36, further comprising:
receiving user input specifying one or more user interface events to which the first node corresponds;
wherein the graphical source code is operable to respond to the one or more user interface events to which the first node corresponds.

42. (Currently Amended) The method of claim 41,
wherein said associating graphical source code with the first node in response to user input comprises associating two or more portions of graphical source code with the first node in response to user input;

wherein each of the portions of graphical source code is operable to respond to one or more of the one or more user interface events.

43. (Currently Amended) The method of claim 41,
wherein said receiving user input specifying one or more user interface events to which the first node corresponds comprises receiving user input specifying a name of each of the user interface events.

44. (Currently Amended) The method of claim 41, further comprising:
displaying a graphical user interface dialog;
wherein said receiving user input specifying the one or more user interface events to which the first node corresponds comprises receiving user input via the graphical user interface dialog to specify the one or more user interface events.

45. (Currently Amended) The method of claim 36, wherein the user interface event comprises one of:

- ~~a user interface event;~~
- a system event;
- a timer event.

46. (Currently Amended) The method of claim 36,
wherein the graphical program has a graphical user interface;
~~wherein the event comprises a user interface event;~~
wherein the user interface event is associated with a first user interface element of the graphical user interface.

47. (Previously Presented) The method of claim 46, wherein the first user interface element comprises one of:
an indicator;
a control;
a menu element;

a window.

48. (Currently Amended) The method of claim 36,
wherein the graphical program ~~[[has]]~~ includes a graphical user interface;
~~wherein the event comprises a user interface event;~~
wherein the user interface event is associated with a user action performed on the graphical user interface.

49. (Currently Amended) The method of claim 36, further comprising:
executing the graphical program;
generating the user interface event during execution of the graphical program;
wherein said executing the graphical program includes executing the graphical source code in response to said generating the user interface event.

50. (Currently Amended) The method of claim 36,
wherein the graphical program ~~[[has]]~~ includes a graphical user interface;
wherein the method further comprises:
 executing the graphical program;
 generating the user interface event during execution of the graphical program, wherein said generating the user interface event comprises generating the user interface event in response to user input to the graphical user interface;
 wherein said executing the graphical program includes executing the graphical source code in response to said generating the user interface event.

51. (Previously Presented) The method of claim 36,
wherein the block diagram of the graphical program comprises a data flow diagram.

52. (Previously Presented) The method of claim 36,
wherein the block diagram comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program.

53. (Currently Amended) A memory medium for creating a graphical program, the memory medium comprising program instructions executable to:

display a first node in a block diagram of the graphical program in response to user input; and

associate graphical source code with the first node in response to user input;

wherein the graphical source code is operable to execute in response to [[an]] a user interface event.

54. (Currently Amended) The memory medium of claim 53,

wherein said graphical source code executing in response to the user interface event comprises the graphical source code executing in response to the user interface event during execution of the graphical program, wherein the user interface event is generated during execution of the graphical program.

55. (Previously Presented) The memory medium of claim 53,

wherein the graphical source code comprises a plurality of interconnected nodes.

56. (Previously Presented) The memory medium of claim 55,

wherein said associating the graphical source code with the first node comprises:

displaying the plurality of nodes in response to user input; and

interconnecting the plurality of nodes in response to user input.

57. (Previously Presented) The memory medium of claim 53,

wherein said associating the graphical source code with the first node in response to user input comprises displaying the graphical source code within the first node in response to user input.

58. (Currently Amended) The memory medium of claim 53,

wherein the memory medium further comprises program instructions executable to receive user input specifying one or more user interface events to which the first node corresponds;

wherein the graphical source code is operable to respond to the one or more user interface events to which the first node corresponds.

59. (Currently Amended) The memory medium of claim 58,
wherein said associating graphical source code with the first node in response to user input comprises associating two or more portions of graphical source code with the first node in response to user input;

wherein each of the portions of graphical source code is operable to respond to one or more of the one or more user interface events.

60. (Currently Amended) The method of claim 58,
wherein the memory medium further comprises program instructions executable to display a graphical user interface dialog;

wherein said receiving user input specifying the one or more user interface events to which the first node corresponds comprises receiving user input via the graphical user interface dialog to specify the one or more user interface events.

61. (Currently Amended) The memory medium of claim 53, wherein the user interface event comprises one of:

~~a user interface event;~~

a system event;

a timer event.

62. (Currently Amended) The memory medium of claim 53,
wherein the graphical program ~~[[has]]~~ includes a graphical user interface;
~~wherein the event comprises a user interface event;~~
wherein the user interface event is associated with a first user interface element of the graphical user interface.

63. (Currently Amended) The memory medium of claim 53,
wherein the graphical program ~~[[has]]~~ includes a graphical user interface;
~~wherein the event comprises a user interface event;~~
wherein the user interface event is associated with a user action performed on the graphical user interface.

64. (Previously Presented) The memory medium of claim 53,
wherein the block diagram of the graphical program comprises a data flow diagram.

65. (Previously Presented) The memory medium of claim 53,
wherein the block diagram comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program.

66. (Currently Amended) A system for creating a graphical program, the system comprising:

a memory storing program instructions;

a processor coupled to the memory; and

a display device;

wherein the processor is operable to execute the program instructions stored in the memory to:

display a first node on the display device in response to user input,
wherein said displaying the first node comprises displaying the first node in a block diagram of the graphical program; and

associate graphical source code with the first node in response to user input;

wherein the graphical source code is operable to execute in response to ~~[[an]]~~ a user interface event.